基于 Node. js 的高性能应用服务平台构建

摘　要：本文以新华社马克思主义新闻观学习竞赛活动为切入点，梳理高性能应用服务平台设计难点，并基于 Node.js 搭建出一套竞赛答题应用服务平台，实现了高性能响应的平台框架，参赛人员 / 组织、活动、题目的分级管理，多维度的统计分析，并通过一系列安全性设计实现了数据安全的保障。平台能够应对高并发的业务应用场景，有效保障了新华社马克思主义新闻观学习竞赛活动的顺利进行，并具备支撑更多类似活动的能力。

关键词：高性能；高并发；应用服务平台；Node.js；CDN；系统安全
中图分类号：TP311　　文献标识码：A
文章编号：1671-0134（2018）10-048-02　　DOI：10.19483/j.cnki.11-4653/n.2018.10.015

文 / 李骞

近期，新华社“马克思主义新闻观手机在线学习竞赛活动”圆满落幕。在 15 天的线上答题期间里，70 家社属单位全员参与，全社共有 12000 余人下载活动 APP 参与答题，参与率接近 100%，营造了良好的学习氛围，有效地将新华社马克思主义新闻观教育向纵深推进。而在这次活动的背后，技术的支撑尤为关键。短短 3 个月的时间，我带领三人的自主研发团队，从无到有，从构想到落地，完成了竞赛答题平台的完全自主研发。正是依托这一平台，才能在短时间内快速构建起整个活动的框架。那么这个平台究竟是通过什么样的技术手段来支撑这种活动的呢？

1. 平台设计难点

既谓之“平台”，我们的初衷当然不是一套一次性使用的后台应用，而是一个能够提供给不同用户开展多次活动反复使用的竞赛答题平台。

我们首先通过与活动主办部门的多次研究讨论，建立了本次马克思主义新闻观学习竞赛活动的业务场景：活动分为数个章节，各章题目随机，以闯关形式为主，依次进行。参与者还可对排名靠前者发起挑战，挑战者、被挑战者均有机会获得积分奖励。另外，答完全部章节的参与者继续自行答题，可反复获得积分奖励，积分不设上限。所有问答题目均可实时显示正确答案供参与者学习。

随后我们结合本次活动的业务场景、需求并结合平台化的构想，综合梳理出了竞赛答题平台设计中的难点，具体如下。

1.1 活动模式多

答题竞赛类活动包括闯关、挑战、练习等多种模式，各模式需要能够进行灵活的配置。

1.2 参与单位、参与人员数量多

一次活动应该既可以针对一个单位又可以针对多个单位，由于每个单位又有众多的下属部门、人员，这就造成活动涉及到的人员和单位（部门）数量相当之多。平台如果没有有效的参与单位（部门）、参与人员的管理功能，将会导致很高的管理成本，并影响活动组织效率。

1.3 高并发访问

存在不同程度的并发访问情况。如较长周期的活动会在活动刚开始和即将结束两个时间点产生较大的并发量等。需保证在有较大并发访问的情况下，系统及系统产生的数据不会发生异常，并且维持高性能运转，及时地返回数据请求结果。

1.4 多维度的活动统计

活动中及活动后的统计数据对于这类活动而言十分重要。对于参与者而言，直观的活动统计是他们确认自身答题情况、排名的重要手段；对于组织者而言，多维度的活动统计往往可以作为判断这次活动是否成功的关键参考。

1.5 灵活快速地响应业务变化

平台运行长久与否，很重要的一方面在于其是否能够适应不断变化的业务需求，并进行快速迭代更新。

1.6 数据安全不得有失

活动中涉及到的用户数据及活动本身的数据都需要有较好的安全保障。对于任何数据非法获取或篡改的操作及各种非法请求都必须严加防范。

2. 平台的设计

在平台的设计过程中，我带领团队成员们进行了充分的前期调研，完成了多次 的头脑风暴，从功能设计、技术选型、部署保障等多方面充分应对设计难点。

2.1 功能设计

平台提供人员管理、组织机构管理、题库管理、活动管理、统计分析、用户认证、API 服务等一系列功能和服务，以支撑不同的答题类活动。

通过活动、关卡等的交互式配置界面，支持多活动、多关卡，活动组织者可以轻松地对活动的参数进行调整。

通过提供分级管理、批量数据导入等方式，减轻组织者的单位、人员管理成本。提供按组织结构、活动关卡等多维度的统计查询功能，方便组织者进行实时的统计。

而 API 服务则给活动终端提供多种 API 接口，使得活动终端可以灵活实现各种用户交互功能。

平台所有功能通过角色、内容权限进行划分，只有具备相应权限的用户才可以进行具体功能操作。

2.2 技术框架

平台采用 B/S 架构。

2.2.1 服务端

其中，服务器端基于 Node.js、MySQL 等技术，采用 Express 框架进行开发，并提供符合 RESTful 规范的 API 接口。

Node.js 技术是由事件驱动、非阻塞式 I/O 的编程模型，可充分提高程序执行效率，降低接口延迟。

表 1　各场景下 Node. js 与 JAVA 的性能比较 [1]

场景	每秒可处理的请求数（无并发）		每秒可处理的请求数（有并发）	
	Node.js	Java	Node.js	Java
计算密集	86	181	379	698
I/O 密集	127	37	350	41
计算、I/O 平衡	61	34	221	45
长阻塞 I/O 请求	—	—	64	6

通过数据比较我们可以看到，除了计算密集型场景，在 I/O 密集、计算 I/O 平衡、长阻塞 I/O 请求等各场景下，每秒可处理的请求数 Node.js 比 Java 要多数倍，在并发情况下，Node.js 性能还可以再有数倍的提升。

根据我们的分析，竞赛答题平台主要是进行用户请求的快速处理，很少涉及到复杂的计算，其就是一个典型的 I/O 密集型应用，为了获得最好的平台性能，我们选择使用 Node.js。

Express 是一个简洁、灵活的 Node.js Web 应用开发框架，可以方便快捷地编写 Web 请求处理逻辑并创建 API 服务，并且由于其模块化的特性，便于进行后续的功能扩展。

数据库采用关系型数据库 MySQL，并根据业务需求设计请求性能及扩展性良好的数据库表，保证数据结构的可扩展性，以应对可能的业务变化。由于关系型数据库本身的关系特性，其适合完成复杂的统计查询操作，可以根据业务需求进行多维度的数据查询统计操作。

2.2.2 浏览器端

浏览器端基于 HTML5、CSS3、JavaScript 等技术，采用 Vue.js 框架和 Element UI 库进行开发，实现简洁、易用、模块化的 UI 界面。Vue.js 框架简洁、轻量化，易于进行快速迭代开发，其组件化的特性使得浏览器端的功能扩展性得到保障。Element UI 库提供多样的 UI 组件，可以给用户提供优雅的交互体验，并使得平台管理界面的开发、组装变得更加快速。

2.3 部署设计

平台使用 CDN（内容分发网络）技术为平台活动静态内容提供内容加速服务。CDN 能够自动判别新华通网站动静态内容，通过智能负载均衡系统将用户的访问定位至最佳节点 [2]，能有效避开网络中的拥塞，实现用户最快访问，改善服务效果，为用户访问活动内容提供快捷、稳定的支撑，同时降低应用服务器的访问负载。下面的两幅图为本次活动 CDN 命中示意图 (深色为命中)，通过 CDN 的使用，保护了源站并有效避免了网络拥塞，从而支撑了来自国内全部 34 个省级行政区和海外 100 多个国家及地区的访问。



图 1　CDN 命中示意图（国内）



图 2　CDN 命中示意图（海外）

平台通过使用负载均衡技术，平衡用户的应用服务器访问请求。同时，支持系统在负载较大的情况下，根据系统瓶颈，横向扩展相应组件，如负载均衡服务器、应用服务器、数据库服务器等。

平台通过定期的系统数据备份，确保平台数据在意外丢失时能够及时恢复。

2.4 安全性设计

平台在安全方面采取了多种安全防护措施，确保系统、数据的安全性。例如：

（1）使用 SSL 安全证书，对网络请求进行加密；

（2）对关键信息进行加密处理，从数据源头上进行安全保护；

（3）对服务器进行定期漏洞检测、基线检查、病毒查杀；

（下转第56页）